DIGITAL DOWNLOAD हिंदी

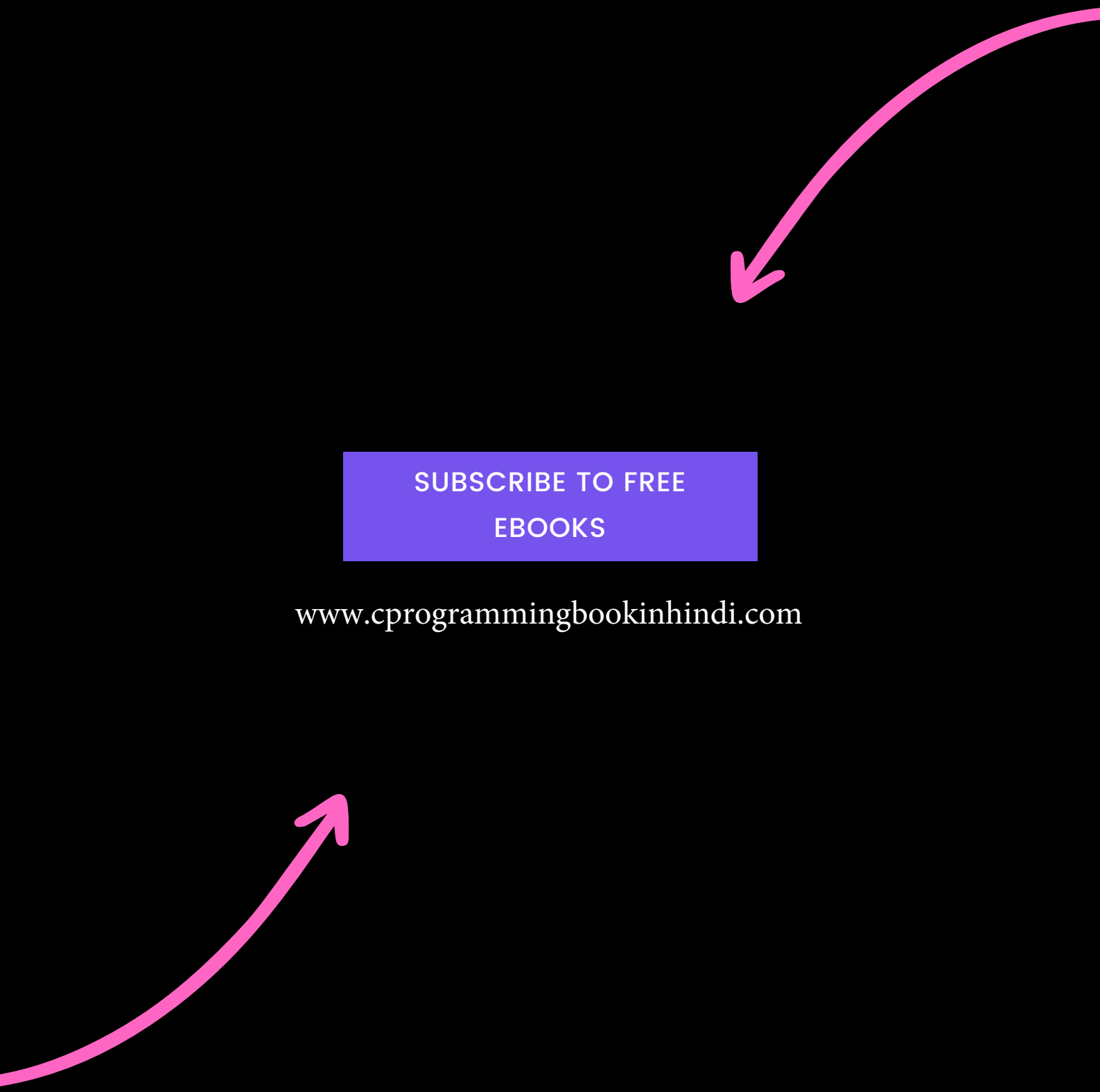
C Programming for Beginners

Thanks for purchasing this Digital Download.

What's the success?

“We learn from failure, not from success!”
— Bram Stoker, Dracula

Please Keep this PDF in a safe place!


SUBSCRIBE TO FREE
EBOOKS
www.cprogrammingbookinhindi.com

C Programming for Beginners in Hindi

Copyright © 2017 by Machhindra Kalan

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Machhindra Kalan

Distributed to the book trade worldwide by Web Study Lab Pvt Ltd, Location - Mumbai, Palava city Pin 421204

E-mail: machhi29@gmail.com, or

Visit: www.cprogrammingbookinhindi.com

For information on translations, please contact Web Study Lab Pvt Ltd, Location - Mumbai, Palava city Pin 421204

Phone +91 86555-67778

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, the author shall not have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this book.

www.cprogrammingbookinhindi.com
3

This book
is dedicated
to those

who really wants
to be a

C Programming for Beginners in Hindi

INDEX OF CONTENTS

Table of Contents

Chapter 1. Getting Ready ……1

Whence C?
Why C?
Whither C?
What Computers Do
High-Level Computer Languages and
Compilers
Using C: Seven Steps
Programming Mechanics
Language Standards
Book Organization
Some Conventions
Summary

Chapter 2. Introducing C …….

A Simple Sample of C
The Explanation
The Structure of a Simple Program
Tips on Making Your Programs Readable
Taking Another Step
While You're at It&"v2" width="21"> Debugging
Keywords and Reserved Identifiers
Key Concepts
Summary

Chapter 3. Data and C …….

A Sample Program
Data Variables and Constants
Data: Data-Type Keywords
C Data Types
Using Data Types
Arguments and Pitfalls
One More Example
Key Concepts

Summary

Chapter 4. Character Strings and Formatted ……..
Input/Output
Introductory Program
Character Strings: An Introduction
Constants and the C Preprocessor
Exploring and Exploiting printf() and
scanf()
Usage Tips
Key Concepts
Summary

Chapter 5. Operators, Expressions, and Statements …….
Introducing Loops
Fundamental Operators
Some Additional Operators
Expressions and Statements
Type Conversions
Function with Arguments
A Sample Program
Key Concepts
Summary

Chapter 6. C Control Statements: Looping ……..
An Initial Example
The while Statement
Which Is Bigger: Using Relational
Operators and Expressions
Indefinite Loops and Counting Loops
The for Loop
More Assignment Operators: +=, -=, *=,/=, %=
The Comma Operator
An Exit-Condition Loop: do while
Which Loop?
Nested Loops
Arrays

A Loop Example Using a Function Return
Value
Key Concepts
Summary

Chapter 7. C Control Statements: Branching and Jumps …….
The if Statement
Adding else to the if Statement
Let's Get Logical
A Word-Count Program
The Conditional Operator:?
Loop Aids: continue and break
Multiple Choice: switch and break
The goto Statement
Key Concepts
Summary

Chapter 8. Character Input/Output and Input Validation …….
Single-Character I/O: getchar() and
putchar()
Buffers
Terminating Keyboard Input
Redirection and Files
Creating a Friendlier User Interface
Input Validation
Menu Browsing
Key Concepts
Summary

Chapter 9. Functions …….
Reviewing Functions
ANSI C Function Prototyping
Recursion
All C Functions Are Created Equal
Compiling Programs with Two or More
Source Code Files
Finding Addresses: The

& Operator
Altering Variables in the Calling Function
Pointers: A First Look
Key Concepts
Summary

Chapter 10. Arrays and Pointers …….
Arrays
Multidimensional Arrays
Pointers and Arrays
Functions, Arrays, and Pointers
Pointer Operations
Protecting Array Contents
Pointers and Multidimensional Arrays
Variable-Length Arrays (VLAs)
Compound Literals
Key Concepts
Summary

CHAPTER 1

GETTING READY

आप इस अध्याय में निम्नलिखित के बारे में जानेंगे:

> C का इतिहास और विशेषताएं
> Program लिखने के लिए आवश्यक कदम
> कंपाइलर और लिंकर्स के बारे में थोड़ा सा
> C standard

C की दुनिया में आपका स्वागत है, शौकिया और वाणिज्यिक प्रोग्रामर के साथ लोकप्रिय पेशेवर प्रोग्रामिंग भाषा। यह अध्याय आपको इस शक्तिशाली और लोकप्रिय भाषा को सीखने और उपयोग करने के लिए तैयार करता है, और यह आपको उन प्रकार के वातावरण से परिचित कराता है जिसमें आप अपने सी–पैरों को विकसित करने की सबसे अधिक रखते हैं।

सबसे पहले, हम सी की उत्पतति को देखते हैं और इसकी कुछ विशेषताओं, दोनों शक्तियों और कमियों की जांच करते हैं। फिर हम प्रोग्रामिंग की उत्पतति को देखते हैं और प्रोग्रामिंग के लिए कुछ सामान्य सिद्धांतों की जांच करते हैं। अंत में, हम चर्चा करते हैं कि कुछ सामान्य प्रणालियों पर C प्रोग्राम कैसे चलाएं।

WHENCE C?

बेल लैब्स के डेनिस रिची ने 1972 में C को बनाया क्योंकि उन्होंने और Ken Thompson ने UNIX ऑपरेटिंग सिस्टम को डिजाइन करने का काम किया।

अधिकांश भाषाएं उपयोगी होने का लक्ष्य रखती हैं, लेकिन उन्हें अक्सर अन्य चिंताएं होती हैं। उदाहरण के लिए, पास्कल का main लक्ष्य अच्छे प्रोग्रामिंग सिद्धांतों को सिखाने के लिए एक अच्छा आधार प्रदान करना था। दूसरी ओर, बेसिक को अंग्रेजी से मिलता–जुलता बनाया गया, ताकि कंप्यूटर से अपरिचित छात्रों द्वारा इसे आसानी से सीखा जा सके। ये महत्वपूर्ण लक्ष्य हैं, लेकिन वे हमेशा व्यावहारिक, काम की उपयोगिता के साथ संगत नहीं हैं। प्रोग्रामर्स के लिए डिज़ाइन की गई भाषा के रूप में सी का विकास, हालांकि, इसे पसंद

की आधुनिक–आधुनिक भाषाओं में से एक बना दिया है।

WHY C?

पिछले तीन दशकों के दौरान, C सबसे महत्वपूर्ण और लोकप्रिय प्रोग्रामिंग भाषाओं में से एक बन गया है। यह बड़ा हो गया है क्योंकि लोग इसे पसंद करते हैं और इसे पसंद करते हैं। पिछले एक दशक में, कई लोग C से अधिक महत्वाकांक्षी C भाषा में चले गए हैं, लेकिन C अभी भी अपने आप में एक महत्वपूर्ण भाषा है, साथ ही C के लिए एक माइग्रेशन पथ भी है। जैसा कि आप C सीखते हैं, आप इसके कई गुणों को पहचानेंगे अब उनमें से कुछ का पूर्वावलोकन करते हैं।

DESIGN FEATURES

C एक आधुनिक भाषा है जिसमें कंप्यूटर विज्ञान के सिद्धांत और अभ्यास द्वारा वांछित नियंत्रण सुविधाओं को शामिल किया गया है। इसका डिज़ाइन उपयोगकर्ताओं के लिए टॉप–डाउन प्लानिंग, संरचित प्रोग्रामिंग और मॉड्यूलर डिज़ाइन का उपयोग करना स्वाभाविक बनाता है। परिणाम एक अधिक विश्वसनीय, समझने योग्य Program है।

EFFICIENCY

सी एक कुशल भाषा है। इसका डिज़ाइन वर्तमान कंप्यूटरों की क्षमताओं का लाभ उठाता है। C प्रोग्राम कॉम्पैक्ट होते हैं और जल्दी से चलते हैं। वास्तव में, सी कुछ ठीक नियंत्रण को प्रदर्शित करता है जो आमतौर पर एक विधानसभा भाषा से जुड़ा होता है। यदि आप चुनते हैं, तो आप अधिकतम गति या स्मृति के सबसे कुशल उपयोग के लिए अपने Programs को ठीक कर सकते हैं।

Portability

C एक पोर्टेबल भाषा है। इसका मतलब है कि एक सिस्टम पर लिखे गए C प्रोग्राम को अन्य सिस्टम पर बहुत कम या बिना किसी संशोधन के चलाया जा सकता है। यदि संशोधन आवश्यक हैं, तो वे अक्सर main Program के साथ एक हेडर फ़ाइल में कुछ प्रविष्टियों को बदलकर बनाया जा सकता है। अधिकांश भाषाओं का अर्थ पोर्टेबल होता है, लेकिन किसी ने भी जो IBM PC BASIC प्रोग्राम को Apple BASIC में बदल दिया है (और वे करीबी

चचेरे भाई हैं) या UNIX सिस्टम पर IBM मेनफ्रेम फोरट्रान प्रोग्राम को चलाने की कोशिश करते हैं, जानते हैं कि पोर्टिंग बिल्कुल भी तकलीफदेह है। C पोर्टेबिलिटी में अग्रणी है। सी कंपाइलर (प्रोग्राम जो आपके सी कोड को कंप्यूटर द्वारा आंतरिक रूप से उपयोग किए जाने वाले निर्देशों में परिवर्तित करते हैं) लगभग 40 प्रणालियों के लिए उपलब्ध हैं, जो 8–बिट माइक्रोप्रोसेसरों से क्रे सुपर कंप्यूटर तक चल रहे हैं। हालाँकि, ध्यान दें कि किसी प्रोग्राम के भाग विशेष रूप से विशेष हार्डवेयर डिवाइसों को एक्सेस करने के लिए लिखे गए हैं, जो कि ऑपरेटिंग सिस्टम के विशेष फीचर्स के रूप में विंडोज या ओएस के रूप में विशेष रूप से पोर्टेबल हैं।

UNIX के साथ C के घनिष्ठ संबंधों के कारण, UNIX सिस्टम आमतौर पर संकुल के भाग के रूप में C compiler के साथ आते हैं। लिनक्स इंस्टॉलेशन में आमतौर पर सी कंपाइलर भी शामिल होता है। पर्सनल कंप्यूटर के लिए कई सी कंपाइलर उपलब्ध हैं, जिनमें विंडोज और मैकिनटोश के विभिन्न संस्करण चलाने वाले पीसी शामिल हैं। तो चाहे आप एक होम कंप्यूटर, एक पेशेवर वर्कस्टेशन, या एक मेनफ्रेम का उपयोग कर रहे हों, अच्छी है कि आप अपने विशेष सिस्टम के लिए सी कंपाइलर प्राप्त कर सकते हैं।

POWER AND FLEXIBILITY

C शक्तिशाली और लचीला है (कंप्यूटर साहित्य में दो पसंदीदा शब्द)। उदाहरण के लिए, अधिकांश शक्तिशाली, लचीली UNIX ऑपरेटिंग सिस्टम C. में लिखी गई है। कई कंप्रेशर्स और दुभाषियों को FORTRAN, APL, पास्कल, LISP, लोगो और BASIC के रूप में C. में लिखा गया है। जब आप एक UNIX मशीन पर FORTRAN का उपयोग करते हैं, तो अंततः अंतिम निष्पादन प्रोग्राम के निर्माण का काम C प्रोग्राम ने किया है। C Programs का उपयोग भौतिकी और इंजीनियरिंग समस्याओं को हल करने के लिए किया गया है और यहां तक कि ग्लेडिएटर जैसी फिल्मों के लिए विशेष प्रभाव को एनिमेट करने के लिए भी।

PROGRAMMER ORIENTED

C प्रोग्रामर्स की जरूरतों को पूरा करने के लिए oriented है। यह आपको हार्डवेयर तक पहुंच प्रदान करता है, और यह आपको व्यक्तिगत बिट्स को मेमोरी में हेरफेर करने में सक्षम बनाता है। इसमें ऑपरेटरों का एक समृद्ध चयन है जो आपको खुद को सफलतापूर्वक व्यक्त करने की अनुमति देता है। सी, पास्कल की तुलना में कम सख्त है, जो आप कर सकते हैं। यह लचीलापन एक फायदा और खतरा दोनों है। लाभ यह है कि कई कार्य, जैसे डेटा के रूप बदलना, सी में बहुत सरल हैं। खतरा यह है कि सी के साथ, आप कुछ भाषाओं में असंभव हैं जो गलतियाँ कर सकते हैं। C आपको अधिक स्वतंत्रता देता है, लेकिन यह आप पर अधिक जिम्मेदारी भी डालता है।

इसके अलावा, अधिकांश सी कार्यान्वयन में उपयोगी सी फ़ंक्शन का एक बड़ा पुस्तकालय है। ये फ़ंक्शन कई जरूरतों से निपटते हैं जो एक प्रोग्रामर आमतौर पर सामना करते हैं।

कमियों

C में कुछ दोष हैं। अक्सर, लोगों के साथ, दोष और गुण समान विशेषता के विपरीत पक्ष होते हैं। उदाहरण के लिए, हमने उल्लेख किया है कि C की अभिव्यक्ति की स्वतंत्रता के लिए अतिरिक्त जिम्मेदारी की भी आवश्यकता है। सी का उपयोग करने वाले पॉइंटर्स (कुछ आप इस पुस्तक के बारे में जानने के लिए तत्पर हो सकते हैं), विशेष रूप से, इसका मतलब है कि आप प्रोग्रामिंग त्रुटियां कर सकते हैं जिन्हें ट्रेस करना बहुत मुश्किल है। जैसा कि एक बार एक कंप्यूटर ने टिप्पणी की थी, स्वतंत्रता की कीमत शाश्वत सतर्कता है।

ऑपरेटरों के अपने धन के साथ संयुक्त सी की सहमति से कोड तैयार करना संभव हो जाता है जिसका पालन करना बेहद मुश्किल है। आप अस्पष्ट कोड लिखने के लिए मजबूर नहीं हैं, लेकिन अवसर है। आखिरकार, अन्य भाषा में एक वार्षिक Obfuscated कोड प्रतियोगिता क्या है?

अधिक गुण हैं और, निस्संदेह, कुछ और दोष हैं। मामले को और आगे बढ़ाने के बजाय, आइए एक नए विषय पर चलते हैं।

सी?

1980 के दशक की शुरुआत तक, सी पहले से ही यूनिक्स प्रणालियों की मिनीकंप्यूटर दुनिया में एक प्रमुख भाषा थी। तब से, यह व्यक्तिगत कंप्यूटर (माइक्रो कंप्यूटर) और मेनफ्रेम (बड़े लोगों) तक फैल गया है। कई सॉफ्टवेयर हाउस वर्ड प्रोसेसिंग प्रोग्राम, स्प्रेडशीट, कंपाइलर और अन्य उत्पादों के उत्पादन के लिए पसंदीदा भाषा के रूप में C का उपयोग करते हैं। ये कंपनियां जानती हैं कि C, कॉम्पैक्ट और कुशल प्रोग्राम तैयार करता है। अधिक महत्वपूर्ण, वे जानते हैं कि इन Programs को संशोधित करना आसान होगा और कंप्यूटर के नए मॉडल के लिए अनुकूल करना आसान होगा।

जहां **C** का उपयोग किया जाता है।

कंपनियों और सी दिग्गजों के लिए क्या अच्छा है अन्य उपयोगकर्ताओं के लिए भी अच्छा है। अधिक से अधिक कंप्यूटर उपयोगकर्ता अपने फायदे के लिए C को चालू कर चुके हैं। C का उपयोग करने के लिए आपके पास कंप्यूटर पेशेवर होना आवश्यक नहीं है

1990 के दशक में, कई सॉफ्टवेयर घरों ने बड़ी प्रोग्रामिंग परियोजनाओं के लिए सी भाषा की ओर रुख करना शुरू कर दिया। C ग्राफ्ट ऑब्जेक्ट–ओरिएंटेड प्रोग्रामिंग टूल्स C भाषा में। C, C का लगभग एक सुपरसेट है, जिसका अर्थ है कि कोई भी C प्रोग्राम है, या लगभग, एक मान्य C प्रोग्राम है। C सीखने से, आप C से बहुत कुछ सीखते हैं।

सी और जावा जैसी नई भाषाओं की लोकप्रियता के बावजूद, सी सॉफ्टवेयर व्यवसाय में एक main कौशल बना हुआ है, आम तौर पर वांछित कौशल के शीर्ष दस में रैंकिंग। विशेष रूप से, सी प्रोग्रामिंग एम्बेडेड सिस्टम के लिए लोकप्रिय हो गया है। यही है, इसका उपयोग ऑटोमोबाइल, कैमरा, डीवीडी प्लेयर और अन्य आधुनिक उपयुक्तताओं में पाए जाने वाले तेजी से सामान्य माइक्रोप्रोसेसरों को प्रोग्राम करने के लिए किया जाता है। इसके अलावा, C, FORTRAN के वैज्ञानिक प्रोग्रामिंग के लंबे प्रभुत्व में अतिक्रमण कर रहा है। अंत में, चूंकि ऑपरेटिंग सिस्टम को विकसित करने के लिए बनाई गई भाषा में, यह लिनक्स के विकास में एक मजबूत भूमिका निभाता है। इस प्रकार, इक्कीसवीं सदी के पहले दशक में सी अभी भी मजबूत हो रहा है।

संक्षेप में, C सबसे महत्वपूर्ण प्रोग्रामिंग भाषाओं में से एक है और ऐसा करना जारी रखेगा। अगर तुम एक नौकरी लेखन सॉफ्टवेयर चाहते हैं, पहले प्रश्नों में से एक जो आपको हां में जवाब देने में सक्षम होना चाहिए "ओह कहते हैं," क्या आप सी? "

कंप्यूटर क्या करते हैं

अब जब आप C में प्रोग्राम करना सीखेंगे, तो आपको कंप्यूटर के काम करने के तरीके के बारे में थोड़ा जानना चाहिए। यह ज्ञान आपको सी में एक Program लिखने के बीच के कनेक्शन को समझने में मदद करेगा और जब आप उस Program को चलाते हैं तो आखिर क्या होता है।

आधुनिक कंप्यूटर में कई घटक होते हैं। केंद्रीय प्रसंस्करण इकाई, या सीपीयू, कंप्यूटिंग के अधिकांश काम करता है। रैंडम–एक्सेस मेमोरी या रैम, प्रोग्राम्स और फाइल्स को होल्ड करने के लिए वर्कस्पेस के रूप में कार्य करता है। स्थायी मेमोरी, आमतौर पर एक हार्ड डिस्क, उन Programs और फाइलों को याद करती है, भले ही कंप्यूटर बंद हो। और कंप्यूटर, आप के बीच संचार के लिए कीबोर्ड, माउस, और मॉनीटरप्रूव के रूप में विभिन्न पेरिफेरोच। सीपीयू आपके Programs को संसाधित करता है, तो चलो इसकी भूमिका पर ध्यान केंद्रित करते हैं।

कम से कम इस साधारण खाते में एक सीपीयू का जीवन काफी सरल है। यह मेमोरी से एक निर्देश प्राप्त करता है और इसे निष्पादित करता है। यह स्मृति से अगला निर्देश प्राप्त करता है और इसे निष्पादित करता है, और इसी तरह। (एक गिगाहर्ट्ज़ सीपीयू एक सेकंड में लगभग एक अरब बार ऐसा कर सकता है, इसलिए सीपीयू अपने बोरिंग जीवन को जबरदस्त गति से आगे बढ़ा सकता है।) सीपीयू का अपना छोटा कार्यक्षेत्र है, जिसमें कई रजिस्टर होते हैं, जिनमें से प्रत्येक एक संख्या को पकड़ सकता है। एक रजिस्टर अगले निर्देश का मेमोरी पता रखता है, और सीपीयू इस जानकारी का उपयोग अगले निर्देश को लाने के लिए करता है। एक निर्देश प्राप्त करने के बाद, सीपीयू निर्देश को दूसरे रजिस्टर में संग्रहीत करता है और पहले रजिस्टर को अगले निर्देश के पते पर अपडेट करता है।

सीपीयू के पास निर्देशों (निर्देश सेट) का सीमित प्रदर्शन होता है जिसे वह समझता है। इसके अलावा, ये निर्देश विशिष्ट हैं; उनमें से कई कंप्यूटर को एक स्थान से दूसरे स्थान पर स्थानांतरित करने के लिए कहते हैं, उदाहरण के लिए, स्मृति स्थान से एक रजिस्टर तक।

कुछ दिलचस्प बिंदु हैं जो इस खाते के साथ चलते हैं। सबसे पहले, एक कंप्यूटर में संग्रहीत सब कुछ एक नंबर के रूप में संग्रहीत किया जाता है। संख्या के रूप में संग्रहीत किए जाते हैं। character, जैसे किसी पाठ दस्तावेज़ में आपके द्वारा उपयोग किए जाने वाले character, संख्या के रूप में संग्रहीत किए जाते हैं; प्रत्येक character का एक संख्यात्मक कोड होता है। निर्देश जो एक कंप्यूटर को अपने रजिस्टरों में लोड करता है उसे संख्याओं के रूप में संग्रहीत किया जाता है; निर्देश सेट में प्रत्येक निर्देश का एक संख्यात्मक कोड होता है। दूसरा, कंप्यूटर प्रोग्रामों को अंततः इस संख्यात्मक निर्देश कोड में व्यक्त किया जाना चाहिए, या जिसे मशीन भाषा कहा जाता है।

कंप्यूटर कैसे काम करता है इसका एक परिणाम यह है कि यदि आप कंप्यूटर को कुछ करना चाहते हैं, तो आपको निर्देशों की एक विशेष सूची (एक Program) खिलाना होगा, जिसमें यह बताया जाए कि क्या करना है और कैसे करना है। आपको प्रोग्राम को ऐसी भाषा में बनाना होगा जिसे कंप्यूटर सीधे (मशीन भाषा) समझता हो। यह एक विस्तृत, थकाऊ, सटीक कार्य है। दो नंबरों को एक साथ जोड़ने के रूप में सरल कुछ variables में टूटना होगा:

1. Copy the number in memory location 2000 to register 1.
2. Copy the number in memory location 2004 to register 2.
3. Add the contents of register 2 to the contents of register 1 leaving the answer in register 1.
4. Copy the contents of register 1 to memory location 2008.

और आपको इन सभी निर्देशों का एक संख्यात्मक कोड के साथ प्रतिनिधित्व करना होगा!

यदि इस तरह से एक Program लिखने से कुछ ऐसा लगता है जिसे आप करना चाहते

हैं, तो आपको यह जानकर दुख होगा कि मशीन–भाषा प्रोग्रामिंग का character युग लंबे समय तक है। लेकिन अगर आप कुछ अधिक पसंद करते हैं, तो अपने दिल को High-Level प्रोग्रामिंग भाषाओं के लिए खोलें।

HIGH-LEVEL COMPUTER LANGUAGES AND COMPILERS

High-Level प्रोग्रामिंग भाषाएं, जैसे C, आपके प्रोग्रामिंग जीवन को कई तरीकों से सरल बनाती हैं। सबसे पहले, आपको अपने निर्देशों को एक संख्यात्मक कोड में व्यक्त करने की आवश्यकता नहीं है। दूसरा, आपके द्वारा उपयोग किए जाने वाले निर्देश बहुत करीब हैं कि आप किसी समस्या के बारे में कैसे सोच सकते हैं, क्योंकि वे कंप्यूटर के विस्तृत दृष्टिकोण के बारे में हैं। एक विशेष कार्य को पूरा करने के लिए एक विशेष सीपीयू के बारे में सटीक कदमों के बारे में चिंता करने के बजाय, आप अपनी इच्छाओं को अधिक सार स्तर पर व्यक्त कर सकते हैं। उदाहरण के लिए, दो नंबर जोड़ने के लिए, आप निम्नलिखित लिख सकते हैं:

```
total = mine + yours;
```

कोड को इस तरह से देखकर, आपके पास एक अच्छा विचार है कि वह क्या करता है; संख्यात्मक कोड में व्यक्त कई निर्देशों के बराबर मशीन–भाषा को देखना बहुत कम ज्ञानवर्धक है।

दुर्भाग्य से, विपरीत एक कंप्यूटर के लिए सच है; इसके लिए, उच्च–स्तरीय निर्देश असाध्य जिबरिश है। यह वह जगह है जहाँ compiler तस्वीर में प्रवेश करते हैं। कंपाइलर एक ऐसा प्रोग्राम है जो High-Level भाषा के प्रोग्राम को मशीन भाषा के विस्तृत सेट में निर्देश देता है जो कंप्यूटर को निर्देश की आवश्यकता होती है। आप उच्च–स्तरीय सोच रखते हैं; compiler थकाऊ विवरण का ख्याल रखता है।

compiler दृष्टिकोण का एक और लाभ है। सामान्य तौर पर, प्रत्येक कंप्यूटर डिजाइन की अपनी अनूठी मशीन भाषा होती है। तो, मशीन के लिए भाषा में लिखा गया एक Program, कहते हैं, एक इंटेल पेंटियम सीपीयू का मतलब मोटोरोला पावरपीसी सीपीयू के लिए कुछ

भी नहीं है। लेकिन आप एक विशेष मशीन भाषा के लिए एक compiler से मेल कर सकते हैं। इस प्रकार, सही compiler या compiler के सेट के साथ, आप एक ही उच्च–स्तरीय भाषा Program को विभिन्न मशीन–भाषा Programs की एक किस्म में परिवर्तित कर सकते हैं। आप एक प्रोग्रामिंग समस्या को एक बार हल करते हैं, और फिर आप अपने compiler को कई प्रकार की मशीन भाषाओं में समाधान का अनुवाद करने देते हैं।

संक्षेप में, उच्च–स्तरीय भाषाएं, जैसे कि सी, जावा और पास्कल, अधिक अमूर्त रूप में क्रियाओं का characterन करती हैं और किसी विशेष सीपीयू या निर्देश सेट से बंधी नहीं होती हैं। इसके अलावा, उच्च–स्तरीय भाषाओं को सीखना आसान है और मशीन भाषाओं की तुलना में प्रोग्राम करना बहुत आसान है।

USING C: SEVEN STEPS

सी, जैसा कि आपने देखा है, एक compiled भाषा है। यदि आप पास्कल या फोरट्रान जैसे compiled भाषा का उपयोग करने के आदी हैं, तो आप एक सी प्रोग्राम को एक साथ रखने के मूल से परिचित होंगे।

हालाँकि, यदि आपकी पृष्ठभूमि किसी व्याख्या की गई भाषा में है, जैसे कि BASI, या ग्राफ़िकल–इंटरफ़ेस oriented भाषा में, जैसे Visual Basic, या यदि आपके पास कोई पृष्ठभूमि नहीं है, तो आपको यह सीखने की आवश्यकता है कि कैसे compiled किया जाए। हम जल्द ही उस प्रक्रिया को देखेंगे, और आप देखेंगे कि यह सीधा और समझदार है। सबसे पहले, आपको प्रोग्रामिंग का अवलोकन देने के लिए, चलो सी प्रोग्राम लिखने के कार्य को सात अधिनियम में तोड़ते हैं। ध्यान दें कि यह एक आदर्शीकरण है। व्यवहार में, विशेष रूप से बड़ी परियोजनाओं के लिए, आप पहले को परिष्कृत करने के लिए बाद में जो कुछ सीखते थे, उसका उपयोग करके आप आगे–पीछे हो जाएंगे।

प्रोग्रामिंग के सात चरण

चरण **1:** प्रोग्राम के उद्देश्यों को परिभाषित करें

स्वाभाविक रूप से पर्याप्त है, आपको एक स्पष्ट विचार के साथ शुरू करना चाहिए कि आप Program क्या करना चाहते हैं। आपके Program की जरूरत की जानकारी के बारे में सोचें, Program के लिए गणना और हेरफेर के कार्यों को करने की जरूरत है, और Program की जानकारी आपको वापस रिपोर्ट करनी चाहिए। नियोजन के इस स्तर पर, आपको सामान्य शब्दों में सोचना चाहिए, न कि कुछ विशिष्ट कंप्यूटर भाषा के संदर्भ में।

चरण **2:** डिजाइन **Program**

आपके पास इस बात की एक वैचारिक तस्वीर है कि आपके Program को क्या करना चाहिए, आपको यह तय करना चाहिए कि Program इसके बारे में कैसे जाएगा। उपयोगकर्ता इंटरफ़ेस कैसा होना चाहिए? Program कैसे आयोजित किया जाना चाहिए? लक्ष्य उपयोगकर्ता कौन होगा? Program को पूरा करने के लिए आपके पास कितना समय है?

आपको यह भी तय करने की आवश्यकता है कि प्रोग्राम में डेटा का प्रतिनिधित्व कैसे किया जाए और, संभवतः, सहायक फ़ाइलों में, और डेटा को संसाधित करने के लिए किन तरीकों का उपयोग किया जाए। जब आप पहली बार C में प्रोग्रामिंग सीखते हैं, तो विकल्प सरल होंगे, लेकिन जब आप अधिक जटिल परिस्थितियों से निपटेंगे, तो आप पाएंगे कि इन निर्णयों के लिए अधिक विचार की आवश्यकता है। जानकारी का प्रतिनिधित्व करने के लिए एक अच्छा तरीका चुनना अक्सर प्रोग्राम को डिजाइन करना और डेटा को बहुत आसान प्रसंस्करण कर सकता है।

फिर से, आपको सामान्य शब्दों में सोचना चाहिए, विशिष्ट कोड के बारे में नहीं, लेकिन आपके कुछ निर्णय भाषा की सामान्य विशेषताओं पर आधारित हो सकते हैं। उदाहरण के लिए, एक सी प्रोग्रामर के पास डेटा प्रतिनिधित्व से अधिक विकल्प हैं, जैसे कि, पास्कल प्रोग्रामर।

चरण **3:** कोड लिखें

अब जब आपके पास अपने Program के लिए एक स्पष्ट डिजाइन है, तो आप कोड लिखकर इसे लागू करना शुरू कर सकते हैं। यही है, आप अपने प्रोग्राम डिज़ाइन का अनुवाद सी भाषा में करते हैं। यहां वह जगह है जहां आपको वास्तव में सी के अपने ज्ञान को काम में लाना है। आप कागज पर अपने विचारों को स्केच कर सकते हैं, लेकिन अंततः आपको अपना कोड कंप्यूटर में लाना होगा। इस प्रक्रिया के यांत्रिकी आपके प्रोग्रामिंग वातावरण पर निर्भर करते हैं। हम जल्द ही कुछ सामान्य वातावरण के लिए विवरण प्रस्तुत करेंगे। सामान्य तौर पर, आप स्रोत कोड फ़ाइल नामक एक पाठ संपादक का उपयोग करते हैं। इस फ़ाइल में आपके प्रोग्राम डिज़ाइन का C प्रतिपादन है। लिस्टिंग1.1 सी स्रोत कोड का एक उदाहरण दिखाता है।

Listing 1.1 Example of C Source Code

```
#include <stdio.h>
int main(void)
{
 int dogs;
 printf("How many dogs do you have?\n");
 scanf("%d", &dogs);
 printf("So you have %d dog(s)!\n", dogs);
 return 0;
}
```

इस कदम के हिस्से के रूप में, आपको अपने काम का दस्तावेजीकरण करना चाहिए। सबसे सरल तरीका है कि आप अपने सोर्स कोड में स्पष्टीकरण को शामिल करने के लिए C की टिप्पणी सुविधा का उपयोग करें। अध्याय 2, "C का परिचय", के बारे में अधिक बताएगा अपने कोड में टिप्पणियों का उपयोग करना।

Step 4: Compile

अगला Step स्रोत कोड को compiled करना है। फिर से, विवरण आपके प्रोग्रामिंग वातावरण पर निर्भर करता है, और हम जल्द ही कुछ सामान्य वातावरणों को देखेंगे। अभी के लिए, आइए एक और वैचारिक दृष्टिकोण के साथ शुरू करते हैं कि क्या होता है।

याद रखें कि कंपाइलर एक प्रोग्राम है जिसका काम सोर्स कोड को एक्जीक्यूटेबल कोड में बदलना है। निष्पादन योग्य कोड आपके कंप्यूटर की मूल, या मशीन, भाषा में कोड होता है। इस भाषा में एक संख्यात्मक कोड में व्यक्त विस्तृत निर्देश शामिल हैं। जैसा कि आपने पहले पढ़ा था, विभिन्न कंप्यूटरों में अलग–अलग मशीन भाषाएं होती हैं, और एक सी कंपाइलर सी को एक विशेष मशीन भाषा में अनुवाद करता है। सी compiler भी अंतिम Program में सी पुस्तकालयों से कोड शामिल करते हैं; पुस्तकालयों में आपके उपयोग के लिए standard रूटीन का एक फंड होता है, जैसे कि printf() और Scanf() । (अधिक सटीक रूप से, एक प्रोग्राम जिसे लिंकर कहा जाता है, लाइब्रेरी रूटीन में लाता है, लेकिन कंपाइलर आपके लिए अधिकांश सिस्टम पर लिंकर चलाता है।) अंतिम परिणाम एक निष्पादन योग्य फ़ाइल है जिसमें कोड होता है जिसे कंप्यूटर समझता है और जिसे आप चला सकते हैं।

कंपाइलर यह भी जाँचता है कि आपका प्रोग्राम C. वैध है। यदि कंपाइलर को त्रुटियां मिलती हैं, तो यह उन्हें आपकी रिपोर्ट करता है और एक निष्पादन योग्य फ़ाइल का उत्पादन नहीं करता है। एक विशेष compiler की शिकायतों को समझना एक और कौशल है जिसे आप उठाएंगे।

Step 5: Run the Program

परंपरागत रूप से, निष्पादन योग्य फ़ाइल एक प्रोग्राम है जिसे आप चला सकते हैं। MS–DOS और UNIX और Linux कंसोल सहित कई सामान्य वातावरण में प्रोग्राम को चलाने के लिए, बस निष्पादन योग्य फ़ाइल का नाम टाइप करें। अन्य वातावरण, जैसे VMS ऑन VAX, के लिए रन कमांड या किसी अन्य तंत्र की आवश्यकता हो सकती है। एकीकृत विकास वातावरण (आईडीई), जैसे कि विंडोज और मैकिंटोश वातावरण के लिए प्रदान किए गए, आपको मेनू से विकल्प चुनकर या विशेष कुंजी दबाकर आईडीई के भीतर से अपने सी Program को संपादित करने और निष्पादित करने की अनुमति देते हैं। परिणामी

Program भी ऑपरेटिंग सिस्टम से सीधे फ़ाइल नाम या आइकन पर डबल–क्लिक करके चलाया जा सकता है।

Step 6: Test and Debug the Program

तथ्य यह है कि आपका Program चलता है एक अच्छा संकेत है, लेकिन यह संभव है कि यह गलत तरीके से चल सकता है। नतीजतन, आपको यह देखने के लिए जांचना चाहिए कि आपका Program वही करता है जो उसे करना चाहिए था। आप पाएंगे कि आपके कुछ प्रोग्रामों में गलतियां हैं, कंप्यूटर शब्दजाल में। डिबगिंग प्रोग्राम त्रुटियों को ढूंढ रहा है और ठीक कर रहा है। गलतियाँ करना सीखने का एक स्वाभाविक हिस्सा है। यह प्रोग्रामिंग के लिए अंतर्निहित लगता है, इसलिए जब आप सीखने और प्रोग्रामिंग को जोड़ते हैं, तो आपने अपने आप को अक्सर अपनी गिरावट की याद दिलाने के लिए तैयार किया था। जैसे–जैसे आप अधिक शक्तिशाली और सूक्ष्म प्रोग्रामर बनेंगे, वैसे–वैसे आपकी त्रुटियां भी अधिक शक्तिशाली और सूक्ष्म होती जाएंगी।

आपके पास गलती करने के कई अवसर हैं। आप एक मूल डिज़ाइन त्रुटि कर सकते हैं। आप अच्छे विचारों को गलत तरीके से लागू कर सकते हैं। आप अनपेक्षित इनपुट को अनदेखा कर सकते हैं जो आपके प्रोग्राम को गड़बड़ कर देता है। आप गलत तरीके से C का उपयोग कर सकते हैं। आप टाइपिंग त्रुटियां कर सकते हैं। आप brackets को गलत जगह पर रख सकते हैं, और इसी तरह। आपको इस सूची में जोड़ने के लिए अपने स्वयं के आइटम मिलेंगे।

सौभाग्य से, स्थिति निराशाजनक नहीं है, हालांकि कई बार ऐसा हो सकता है जब आपको लगता है कि यह है। कंपाइलर कई प्रकार की त्रुटियों को पकड़ता है, और ऐसी चीजें हैं जो आप अपने आप को ट्रैक करने में मदद करने के लिए कर सकते हैं जो कंपाइलर नहीं पकड़ता है। यह पुस्तक आपको डिबगिंग सलाह देगी जैसा कि आप साथ चलते हैं।

Step 7: Maintain and Modify the Program

जब आप अपने लिए या किसी और के लिए एक Program बनाते हैं, तो उस Program का व्यापक उपयोग हो सकता है। यदि ऐसा होता है, तो संभवतः आपको इसमें बदलाव करने के कारण मिलेंगे। शायद एक मामूली बग है जो केवल तभी दिखाई देता है जब कोई व्यक्ति Zz से शुरू होने वाले नाम में प्रवेश करता है, या आप प्रोग्राम में कुछ करने के लिए एक बेहतर तरीका सोच सकते हैं। आप एक चतुर नई सुविधा जोड़ सकते हैं। आप प्रोग्राम को अनुकूलित कर सकते हैं ताकि यह एक अलग कंप्यूटर सिस्टम पर चले। यदि आप Program को स्पष्ट रूप से दस्तावेज करते हैं और यदि आप ध्वनि डिजाइन प्रथाओं का पालन करते हैं तो ये सभी कार्य बहुत सरल हो जाते हैं।

COMMENTARY

प्रोग्रामिंग आमतौर पर रैखिक के रूप में नहीं है जैसा कि प्रक्रिया केवल वर्णित है। कभी–कभी आपको कदमो के बीच आगे और पीछे जाना पड़ता है। उदाहरण के लिए, जब आप कोड लिख रहे होते हैं, तो आप पाते हैं कि आपकी योजना अव्यावहारिक थी। आप चीजों को करने का एक बेहतर तरीका देख सकते हैं या, यह देखने के बाद कि कोई प्रोग्राम कैसे चलता है, आप डिजाइन को बदलने के लिए प्रेरित महसूस कर सकते हैं। अपने काम का दस्तावेजीकरण आपको स्तरों के बीच आगे और पीछे जाने में मदद करता है।

अधिकांश शिक्षार्थी चरण 1 और 2 की उपेक्षा करते हैं (Program के उद्देश्यों को परिभाषित करना और Program को डिजाइन करना) और सीधे चरण 3 (Program लिखना) पर जाते हैं। आपके द्वारा लिखे गए पहले प्रोग्राम काफी सरल हैं जिन्हें आप अपने सिर में पूरी प्रक्रिया की कल्पना कर सकते हैं। यदि आप कोई गलती करते हैं, तो इसे खोजना आसान है। जैसे–जैसे आपके Program लंबे और अधिक जटिल होते जाते हैं, मानसिक दृश्य विफल होने लगते हैं, और त्रुटियों को ढूंढना मुश्किल हो जाता है। आखिरकार, नियोजन के कदमों की उपेक्षा करने वालों की हार के समय, भ्रम, और हताशा के घंटों की निंदा की जाती है क्योंकि वे बदसूरत, दुविधापूर्ण और अपमानजनक Program बनाते हैं। नौकरी जितनी बड़ी और जटिल होती है, उतनी ही अधिक योजना की आवश्यकता होती है।

यहां नैतिक यह है कि आपको कोडिंग से पहले योजना बनाने की आदत विकसित करनी

चाहिए। अपने Program के उद्देश्यों को रेखांकित करने और डिजाइन की रूपरेखा तैयार करने के लिए प्राचीन लेकिन सम्मानजनक पेन–एंड–पेंसिल तकनीक का उपयोग करें। यदि आप ऐसा करते हैं, तो आप अंततः बचाए गए और संतुष्टि प्राप्त समय में पर्याप्त लाभांश प्राप्त करेंगे।

प्रोग्रामिंग मैकेनिक्स

प्रोग्राम का निर्माण करने के लिए आपको जिन सटीक चरणों का पालन करना चाहिए, वे आपके कंप्यूटर वातावरण पर निर्भर करते हैं। क्योंकि C पोर्टेबल है, यह कई वातावरणों में उपलब्ध है, जिसमें UNIX, Linux, MS–DOS (हाँ, कुछ लोग अभी भी इसका उपयोग करते हैं), Windows और Macintosh OS शामिल हैं। इस पुस्तक में सभी वातावरणों को कवर करने के लिए पर्याप्त जगह नहीं है, खासकर क्योंकि विशेष उत्पाद विकसित होते हैं, मर जाते हैं, और प्रतिस्थापित हो जाते हैं।

पहले, हालांकि, कई सी वातावरणों द्वारा साझा किए गए कुछ पहलुओं पर नजर डालते हैं, जिनमें पांच हमारे द्वारा उल्लिखित हैं। आपको वास्तव में यह जानने की आवश्यकता नहीं है कि सी प्रोग्राम को चलाने के लिए क्या होता है, लेकिन यह अच्छी पृष्ठभूमि है। यह आपको यह समझने में भी मदद कर सकता है कि आपको सी प्रोग्राम प्राप्त करने के लिए कुछ विशेष चरणों से क्यों गुजरना पड़ता है।

जब आप सी भाषा में एक Program लिखते हैं, तो आप एक सोर्स कोड फ़ाइल नामक एक टेक्स्ट फ़ाइल में जो लिखते हैं उसे स्टोर करते हैं। अधिकांश सी सिस्टम, जिनमें हमने उल्लेख किया है, के लिए आवश्यक है कि फ़ाइल का नाम समाप्त हो

.c: उदाहरण के लिए, wordcount.c और budget.c पीरियड से पहले के नाम के हिस्से को बेसन कहा जाता है और अवधि के बाद के हिस्से को एक्सटेंशन कहा जाता है। इसलिए, बजट एक आधार है और सी एक extension है। संयोजन budget.c फ़ाइल नाम है। नाम को विशेष रूप से कंप्यूटर ऑपरेटिंग सिस्टम की आवश्यकताओं को पूरा करना चाहिए। उदाहरण के लिए, MS–DOS IBM PC और क्लोन के लिए एक ऑपरेटिंग सिस्टम है। यह आवश्यक है कि बेसन 8 characters से अधिक लंबा न हो, इसलिए शब्दकोष।पहले

उल्लिखित सी नाम एक वैध डॉस फ़ाइल नाम नहीं होगा। कुछ UNIX सिस्टम extension सहित पूरे नाम पर 14–character सीमा रखते हैं; अन्य UNIX सिस्टम 255 characters तक लंबे नामों की अनुमति देते हैं। लिनक्स, विंडोज और मैकिंटोश ओएस भी लंबे नामों की अनुमति देते हैं।

ताकि हमें संदर्भित करने के लिए कुछ ठोस हो, मान लें कि हमारे पास एक स्रोत फ़ाइल है जिसे कहा जाता है concrete.c लिस्टिंग में सी स्रोत कोड युक्त 1.2.

Listing 1.2 The concrete.c Program

```
#include <stdio.h>
int main(void)
{
 printf("Concrete contains gravel and cement.\n");
 return 0;
}
```

लिस्टिंग 1.2 में दिखाए गए सोर्स कोड फ़ाइल के विवरण के बारे में चिंता न करें आप उनके बारे में अध्याय 2 में जानेंगे.

OBJECT CODE FILES, EXECUTABLE FILES, AND LIBRARIES

सी प्रोग्रामिंग में मूल रणनीति उन Programs का उपयोग करना है जो आपकी स्रोत कोड फ़ाइल को एक निष्पादन योग्य फ़ाइल में परिवर्तित करते हैं, जो एक फाइल है जिसमें रेडी–टू–रन मशीन भाषा कोड है। C क्रियान्वयन करते हैं

यह दो चरणों में है: संकलन और लिंकिंग। compiler आपके स्रोत कोड को एक मध्यवर्ती कोड में परिवर्तित करता है, और लिंकर निष्पादन योग्य फ़ाइल का उत्पादन करने के लिए अन्य कोड के साथ इसे जोड़ता है। C Programs के मॉडर्नाइजेशन को सुविधाजनक बनाने के लिए इस दो–भाग दृष्टिकोण का उपयोग करता है। आप अलग–अलग मॉड्यूल को अलग–अलग compiled कर सकते हैं और बाद में compiled मॉड्यूल को संयोजित करने के लिए लिंकर का उपयोग कर सकते हैं। इस प्रकार, यदि आपको एक मॉड्यूल को बदलने

की आवश्यकता है, तो आपको अन्य लोगों को फिर से स्थापित करने की आवश्यकता नहीं है। इसके अलावा, लिंकर आपके प्रोग्राम को प्री–कैप्ड लाइब्रेरी कोड के साथ जोड़ता है।

मध्यवर्ती फ़ाइलों के रूप के लिए कई विकल्प हैं। सबसे प्रचलित विकल्प, और यहां वर्णित कार्यान्वयन द्वारा लिया गया है, स्रोत कोड को मशीन भाषा कोड में परिवर्तित करना है, जिसके परिणामस्वरूप ऑब्जेक्ट कोड फ़ाइल या ऑब्जेक्ट फ़ाइल शॉर्ट में है। (यह मानता है कि आपके स्रोत कोड में एक एकल फ़ाइल है।) हालाँकि ऑब्जेक्ट फ़ाइल में मशीन भाषा कोड है, यह चलाने के लिए तैयार नहीं है। ऑब्जेक्ट फ़ाइल में आपके स्रोत कोड का अनुवाद शामिल है, लेकिन यह अभी तक एक पूर्ण Program नहीं है।

ऑब्जेक्ट कोड फ़ाइल से गायब पहला element स्टार्ट–अप कोड कहा जाता है, जो कोड है जो आपके प्रोग्राम और ऑपरेटिंग सिस्टम के बीच एक इंटरफ़ेस के रूप में कार्य करता है। उदाहरण के लिए, आप DOS के तहत या लिनक्स के तहत एक IBM PC को चला सकते हैं। हार्डवेयर दोनों ही मामलों में समान है, इसलिए एक ही ऑब्जेक्ट कोड दोनों के साथ काम करेगा, लेकिन आपको लिनक्स के लिए DOS के लिए अलग–अलग स्टार्ट–अप कोड की आवश्यकता होगी क्योंकि ये सिस्टम प्रोग्राम को एक दूसरे से अलग तरीके से संभालते हैं।

दूसरा गायब element (element) लाइब्रेरी रूटीन के लिए कोड है। लगभग सभी सी प्रोग्राम रूटीन (जिन्हें फ़ंक्शन कहा जाता है) का उपयोग करते हैं जो standard सी लाइब्रेरी का हिस्सा हैं। उदाहरण के लिए, concrete.c फंक्शन printf() का उपयोग करता है। ऑब्जेक्ट कोड फ़ाइल में इस फ़ंक्शन के लिए कोड नहीं है; इसमें केवल printf() फ़ंक्शन का उपयोग करने के लिए निर्देश शामिल हैं। वास्तविक कोड को किसी अन्य फ़ाइल में संग्रहीत किया जाता है, जिसे लाइब्रेरी कहा जाता है। एक लाइब्रेरी फ़ाइल में कई कार्यों के लिए ऑब्जेक्ट कोड होता है।

लिंकर की भूमिका इन तीन एलिमेंटसॉयर ऑब्जेक्ट कोड, आपके सिस्टम के लिए standard स्टार्ट–अप कोड, और लाइब्रेरी कोड–एंड को एक साथ लाने के लिए है, निष्पादन योग्य

फ़ाइल। लाइब्रेरी कोड के लिए, लिंकर केवल लाइब्रेरी से आपके द्वारा उपयोग किए जाने वाले कार्यों के लिए आवश्यक कोड निकालता है।

संक्षेप में, एक ऑब्जेक्ट फ़ाइल और एक निष्पादन योग्य फ़ाइल दोनों में मशीन भाषा निर्देश शामिल हैं। हालाँकि, ऑब्जेक्ट फ़ाइल में आपके द्वारा उपयोग किए गए कोड के लिए केवल मशीन भाषा अनुवाद होता है, लेकिन निष्पादन योग्य फ़ाइल में आपके द्वारा उपयोग किए जाने वाले लाइब्रेरी रूटीन और स्टार्ट–अप कोड के लिए मशीन कोड भी होता है।

कुछ सिस्टम पर, आपको कंपाइल और लिंक प्रोग्राम अलग से चलाने चाहिए। अन्य प्रणालियों पर, compiler स्वचालित रूप से लिंकर को शुरू करता है, इसलिए आपको केवल संकलन कमांड देना होगा।

अब कुछ विशिष्ट प्रणालियों को देखते हैं।

यूनिक्स प्रणाली

क्योंकि सी की लोकप्रियता UNIX सिस्टम पर शुरू हुई थी, हम वहां शुरू करेंगे।

एक **UNIX** सिस्टम पर संपादन

UNIX C का अपना संपादक नहीं है। इसके बजाय, आप सामान्य–उद्देश्य वाले UNIX संपादकों में से एक का उपयोग करते हैं, जैसे कि emacs, jove, vi या एक X–Windows संपादक।

आपकी दो main जिम्मेदारियां प्रोग्राम को सही ढंग से टाइप कर रही हैं और उस फाइल के लिए एक नाम चुन रही हैं जो प्रोग्राम को स्टोर करेगा। जैसा कि चर्चा है, नाम के साथ समाप्त होना चाहिए।सी। ध्यान दें कि UNIX अपरकेस और लोअरकेस के बीच अंतर करता है। इसलिए, बजट।सी, budget.c, और budget.c, C स्रोत फ़ाइलों के लिए तीन अलग और मान्य नाम हैं, लेकिन BUDGET.C एक मान्य नाम नहीं है क्योंकि यह लोअरकेस c के बजाय

अपरकेस C का उपयोग करता है।

Vi संपादक का उपयोग करते हुए, हमने निम्नलिखित Program तैयार किया और इसे एक फाइल में संग्रहीत किया, जिसे सूचित किया गया inform.c

```
#include
int main(void)
{
 printf("A .c is used to end a C program filename.\n");
 return 0;
```

यह पाठ स्रोत कोड है, और inform.c करें सोर्स फाइल है। यहां महत्वपूर्ण बिंदु यह है कि स्रोत फ़ाइल एक प्रक्रिया की शुरुआत है, न कि अंत।

एक **UNIX** सिस्टम पर संकलन

हमारा Program, हालांकि निर्विवाद रूप से शानदार है, फिर भी कंप्यूटर के लिए अस्पष्ट है। एक कंप्यूटर #includd या printf जैसी चीजों को नहीं समझता है (इस बिंदु पर, आप शायद या तो नहीं करते हैं, लेकिन आप जल्द ही सीखेंगे, जबकि कंप्यूटर नहीं होगा।) जैसा कि हमने पहले चर्चा की थी, हमें अपने कोड (स्रोत कोड) को कंप्यूटर के कोड (मशीन कोड) में अनुवाद करने के लिए एक कंपाइलर की मदद चाहिए। इन प्रयासों का परिणाम निष्पादन योग्य फ़ाइल होगा, जिसमें सभी मशीन कोड होते हैं जो कंप्यूटर को काम करने की आवश्यकता होती है।

UNIX C कंपाइलर को cc कहा जाता है। सूचना compiled करने के लिए inform.c प्रोग्राम, आपको निम्न टाइप करने की आवश्यकता है:

```
cc inform.c
```

कुछ सेकंड के बाद, यूनिक्स प्रॉम्प्ट आपको बताएगा कि विलेख किया गया है। यदि आप प्रोग्राम को ठीक से लिखने में विफल रहे तो आपको चेतावनी और त्रुटि संदेश मिल सकते

हैं, लेकिन मान लें कि आपने सब कुछ ठीक किया। (यदि कंपाइलर (void) शून्य शब्द के बारे में शिकायत करता है, तो आपके सिस्टम ने अभी तक एएनएसआई सी कंपाइलर को अपडेट नहीं किया है। हम जल्द ही standards के बारे में बात करेंगे। इस बीच, उदाहरण से शून्य शब्द हटा दें।) यदि आप अपनी फ़ाइलों को सूचीबद्ध करने के लिए ls का उपयोग करते हैं, तो आप पाएंगे कि एक नई फ़ाइल है जिसे a.out कहा जाता है। यह प्रोग्राम के अनुवाद (या संकलन) वाली निष्पादन योग्य फ़ाइल है।
इसे चलाने के लिए, बस टाइप करें

a.out

UNIX का उपयोग करके C प्रोग्राम तैयार करना।

और ज्ञान आगे बढ़ता है:

A .c is used to end a C program filename.

यदि आप निष्पादन योग्य फ़ाइल रखना चाहते हैं (ए।बाहर), आपको इसका नाम बदलना चाहिए। अन्यथा, फ़ाइल को नए ए से बदल दिया जाता है।अगली बार जब आप एक Program compiled करें।

ऑब्जेक्ट कोड के बारे में क्या? Cc compiler एक ऑब्जेक्ट कोड फाइल बनाता है जिसमें स्रोत कोड के समान बेसनेम होता है, लेकिन ओ एक्सटेंशन के साथ। हमारे उदाहरण में, ऑब्जेक्ट कोड फ़ाइल को सूचना कहा जाता है।ओ, लेकिन आपको यह नहीं मिलेगा, क्योंकि निष्पादन योग्य Program पूरा हो जाने के बाद लिंकर इसे हटा देता है।
हालाँकि, यदि मूल प्रोग्राम एक से अधिक स्रोत कोड फ़ाइल का उपयोग करता है, तो ऑब्जेक्ट कोड फ़ाइलों को सहेजा जाएगा। जब हम पाठ में बाद में कई–फ़ाइल Programs पर चर्चा करते हैं, तो आप देखेंगे कि यह एक अच्छा विचार है।

लिनक्स सिस्टम

लिनक्स एक लोकप्रिय ओपन–सोर्स, UNIX– जैसा ऑपरेटिंग सिस्टम है जो कई तरह के प्लेटफ़ॉर्म पर चलता है, जिसमें IBM कॉमपिटिबल्स और Macintoshes शामिल हैं। लिनक्स पर C प्रोग्राम तैयार करना UNIX सिस्टम के लिए बहुत समान है, सिवाय इसके कि आप सार्वजनिक डोमेन C कंपाइलर का उपयोग करेंगे, जिसे gcc कहा जाता है, जो GNU द्वारा प्रदान किया जाता है। संकलन कमांड निम्न की तरह दिखेगा:

```
gcc inform.c
```

ध्यान दें कि लिनक्स स्थापित करते समय gcc स्थापित करना वैकल्पिक हो सकता है, इसलिए आपको (या किसी को) cc स्थापित करना पड़ सकता है यदि वह पहले स्थापित नहीं था। आमतौर पर, इंस्टॉलेशन ccc को gcc के लिए एक उपनाम बनाता है, इसलिए आप चाहें तो gcc के बजाय कमांड लाइन में cc का उपयोग कर सकते हैं।

आप gcc के बारे में और जानकारी प्राप्त कर सकते हैं, जिसमें नई रिलीज़ के बारे में जानकारी http://www.gnu.org/software/gcc/gcc.html.

INTEGRATED DEVELOPMENT ENVIRONMENTS (WINDOWS)

सी कंपाइलर standard विंडोज पैकेज का हिस्सा नहीं हैं, इसलिए आपको सी कंपाइलर प्राप्त करने और स्थापित करने की आवश्यकता हो सकती है। Microsoft, Borland, और Metrowerks सहित कुछ विक्रेताओं, Windows– आधारित एकीकृत विकास वातावरण, या IDEs प्रदान करते हैं। (इन दिनों, अधिकांश सी और सी compiler संयुक्त हैं।) सभी में C प्रोग्राम्स को एक साथ रखने के लिए तेज, एकीकृत वातावरण है। main बिंदु यह है कि इन Programs में से प्रत्येक में एक अंतर्निहित संपादक है जिसका उपयोग आप सी प्रोग्राम लिखने के लिए कर सकते हैं। प्रत्येक मेनू प्रदान करता है जो आपको अपने स्रोत कोड फ़ाइल को नाम देने और सहेजने में सक्षम बनाता है, साथ ही मेनू जो आपको आईडीई छोड़ने के बिना अपने Program को compiled करने और चलाने की अनुमति देता है। यदि कंपाइलर कोई त्रुटि पाता है, और प्रत्येक आपततिजनक लाइनों की पहचान करता है और

उचित त्रुटि संदेशों से उनका मिलान करता है, तो प्रत्येक आपको संपादक में वापस भेज देता है।

विंडोज आईडीई पहली बार में थोड़ा डराने वाला हो सकता है क्योंकि वे विभिन्न प्रकार के लक्ष्य प्रदान करते हैं, अर्थात विभिन्न प्रकार के वातावरण जिसमें Program का उपयोग किया जाएगा। उदाहरण के लिए, वे आपको 16–बिट विंडोज प्रोग्राम, 32–बिट विंडोज प्रोग्राम, डायनामिक–लिंक लाइब्रेरी फाइल्स (DLL), और इसी तरह का विकल्प दे सकते हैं। लक्ष्य के कई विंडोज ग्राफिकल इंटरफ़ेस के लिए समर्थन में लाना शामिल है। इन (और अन्य) विकल्पों का प्रबंधन करने के लिए, आप आम तौर पर एक प्रोजेक्ट बनाते हैं जिसमें आप तब स्रोत कोड फ़ाइलों के नाम जोड़ते हैं जिनका आप उपयोग कर रहे हैं। सटीक चरण आपके द्वारा उपयोग किए जाने वाले उत्पाद पर निर्भर करते हैं। आमतौर पर, आप प्रोजेक्ट बनाने के लिए पहले फ़ाइल मेनू या प्रोजेक्ट मेनू का उपयोग करते हैं। क्या महत्वपूर्ण है परियोजना का सही रूप चुनना।

इस पुस्तक के उदाहरण एक सामान्य कमांड–लाइन वातावरण में चलाने के लिए डिज़ाइन किए गए सामान्य उदाहरण हैं। विभिन्न विंडोज आईडीई इस निस्संदेह धारणा से मेल खाने के लिए एक या अधिक विकल्प प्रदान करते हैं। माइक्रोसॉफ्ट विजुअल सी6.0, उदाहरण के लिए, Win32 कंसोल एपलिकेशन विकल्प प्रदान करता है। Metrowerks CodeWarrior 6.0 के लिए, Win32 C स्टेशनरी चुनें, और फिर C कंसोल ऐप या WinSIOUX C ऐप चुनें (बाद वाले के पास एक अच्छा इंटरफ़ेस इंटरफ़ेस है)। अन्य प्रणालियों के लिए, OS EXE, कंसोल या कैरेक्टर मोड निष्पादन योग्य जैसे शब्दों का उपयोग करके एक विकल्प की तलाश करें। ये मोड आपके निष्पादन योग्य प्रोग्राम को कंसोल जैसी विंडो में चलाएंगे। आपके पास सही प्रोजेक्ट प्रकार होने के बाद, एक नया स्रोत कोड फ़ाइल खोलने के लिए IDE मेनू का उपयोग करें। अधिकांश उत्पादों के लिए, आप फ़ाइल मेनू का उपयोग करके ऐसा कर सकते हैं। प्रोजेक्ट में स्रोत फ़ाइल जोड़ने के लिए आपको अतिरिक्त कदम उठाने पड़ सकते हैं।

क्योंकि Windows sDE आमतौर पर C और C दोनों को संभालता है, इसलिए आपको संकेत

देना होगा कि आप C प्रोग्राम चाहते हैं। कुछ उत्पादों के साथ, जैसे कि Metrowerks CodeWarri, आप प्रोजेक्ट प्रकार का उपयोग यह इंगित करने के लिए करते हैं कि आप C. का उपयोग करना चाहते हैं। अन्य उत्पादों, जैसे Microsoft Visual C, के साथ आप का उपयोग करते हैं .c फ़ाइल एक्सटेंशन यह इंगित करने के लिए कि आप C के बजाय C का उपयोग करना चाहते हैं। हालाँकि, अधिकांश C प्रोग्राम C प्रोग्राम्स के रूप में भी काम करते हैं। Reference Section IX, "Diferenc Between C and C++," compares C and C++.

एक समस्या जिसका आप सामना कर सकते हैं, वह यह है कि प्रोग्राम के निष्पादन के समय प्रदर्शित होने वाली विंडो गायब हो जाती है। यदि आपके लिए यही स्थिति है, तो जब तक आप Eer कुंजी दबाते हैं, तब तक आप प्रोग्राम को रोक सकते हैं। ऐसा करने के लिए, वापसी कथन से ठीक पहले, Program के अंत में निम्नलिखित पंक्ति जोड़ें:

```
getchar();
```

यह पंक्ति एक कीस्ट्रोक पढ़ती है, इसलिए जब तक आप Enter कुंजी दबाते हैं, तब तक Program रुक जाएगा। कभी–कभी, प्रोग्राम कैसे कार्य करता है, इसके आधार पर, पहले से ही एक कीस्ट्रोके प्रतीक्षा हो सकती है। उस स्थिति में, आपको दो बार गेटचेयर () का उपयोग करना होगा:

```
getchar();
getchar();
```

उदाहरण के लिए, यदि प्रोग्राम ने आखिरी चीज आपसे अपना वजन दर्ज करने के लिए कहा है, तो आपने अपना वजन टाइप किया होगा और फिर डेटा दर्ज करने के लिए Enter कुंजी दबाया होगा। Program भार को पढ़ेगा, पहले getchar () Enter कुंजी को पढ़ेगा, और दूसरा getchar () तब तक प्रोग्राम को रोक देगा, जब तक आप फिर से Enter दबाते हैं। यदि यह आपके लिए बहुत मायने नहीं रखता है, तो यह आपके द्वारा C इनपुट के बारे में अधिक जानने के बाद होगा।

यद्यपि विभिन्न आईडीई कई व्यापक सिद्धांतों को साझा करते हैं, लेकिन विवरण उत्पाद से उत्पाद तक और उत्पाद लाइन के भीतर, संस्करण से संस्करण में भिन्न होते हैं। आपको यह जानने के लिए कुछ प्रयोग करने होंगे कि आपका कंपाइलर कैसे काम करता है। आपको मैनुअल पढ़ना या ऑनलाइन ट्यूटोरियल आज़माना पड़ सकता है।

आईबीएम पीसी के लिए डॉस कंपाइलर

कई लोगों के लिए, एक पीसी पर DOS चलाना इन दिनों फैशन से बाहर है, लेकिन यह अभी भी सीमित कंप्यूटर संसाधनों और मामूली बजट वालों के लिए एक विकल्प है और उन लोगों के लिए जो घंटियाँ, सीटी और बिना विचलित के एक सरल ऑपरेटिंग सिस्टम पसंद करते हैं। खिड़की का माहौल। कई विंडोज आईडीई अतिरिक्त कमांड लाइन टूल प्रदान करते हैं जो आपको डॉस कमांड लाइन वातावरण में प्रोग्राम करने की अनुमति देते हैं। कई सिस्टम पर उपलब्ध कॉम्पे सी / सी कंपाइलर, जिसमें कई यूनिक्स और लिनक्स वेरिएंट शामिल हैं, में कमांड–लाइन डॉस संस्करण है। इसके अलावा, फ्रीवेयर और शेयरवेयर सी कंपाइलर हैं जो डॉस के तहत काम करते हैं। उदाहरण के लिए, GNU gcc compiler का DOS– आधारित संस्करण है।

सोर्स कोड फाइलें टेक्स्ट फाइल होनी चाहिए, वर्ड प्रोसेसर फाइलें नहीं। (वर्ड प्रोसेसर फ़ाइलों में फोंट और फॉर्मेटिंग के बारे में बहुत सारी अतिरिक्त जानकारी होती है।) आपको एक पाठ संपादक का उपयोग करना चाहिए, जैसे कि विंडोज नोटपैड, ईडीआईटी Program जो डॉस के कुछ संस्करणों के साथ आता है, या आप एक वर्ड प्रोसेसर का उपयोग कर सकते हैं, यदि आप टेक्स्ट मोड में फाइल को बचाने के लिए सेव एज़ चरण का उपयोग करते हैं। फ़ाइल एक होना चाहिए .c extension। कुछ शब्द प्रोसेसर स्वचालित रूप से जोड़ते हैं।पाठ फ़ाइलों के लिए .txt एक्सटेंशन। यदि आपके साथ ऐसा होता है, तो आपको c के साथ txt की जगह फ़ाइल नाम बदलना होगा।

पीसी के लिए सी कंपाइलर आमतौर पर, लेकिन हमेशा नहीं, मध्यवर्ती ऑब्जेक्ट कोड फ़ाइलों का उत्पादन करते हैं .obj extension। UNIX कंपाइलर के विपरीत, C कंपाइलर आमतौर

पर इन फ़ाइलों को हटाए जाने पर नहीं हटाते हैं। कुछ कंपाइलर असेंबली लैंग्वेज फाइल तैयार करते हैं।एएसएम एक्सटेंशन या अपने स्वयं के कुछ विशेष प्रारूप का उपयोग करते हैं।

कुछ compiler संकलन के बाद लिंकर को स्वचालित रूप से चलाते हैं; दूसरों को आप मैन्युअल रूप से लिंकर चलाने की आवश्यकता हो सकती है। निष्पादन योग्य फ़ाइल में परिणाम जोड़ना, जो एपेंड करता है।मूल स्रोत कोड बेसनेम के लिए EXE एक्सटेंशन। उदाहरण के लिए, concrete.c नामक स्रोत कोड फ़ाइल को compiled करना और लिंक करना। एक फाइल बनाता है जिसे concrete.c प्रोग्राम फ़ाइल कहा जाता है। कुछ compiler एक निष्पादन योग्य नाम कंक्रीट बनाने का विकल्प प्रदान करते हैं।इसके बजाय concrete.com। किसी भी स्थिति में, आप कमांड लाइन पर बेसन टाइप करके प्रोग्राम चला सकते हैं:

C>concrete

C ON THE MACINTOSH

सबसे प्रसिद्ध Macintosh C++ compiler Metrowerks CodeWarrior compiler है। (कोडवेयर के विंडोज और मैकिंटोश संस्करणों में बहुत समान इंटरफेस हैं।) यह एक परियोजना–आधारित आईडीई प्रदान करता है जो आपको विंडोज कंपाइलर में मिलता है। फ़ाइल मेनू से नया प्रोजेक्ट चुनकर start करें। आपको प्रोजेक्ट प्रकारों का विकल्प दिया जाएगा। हाल के कोडवर्ड संस्करणों के लिए, Std C कंसोल विकल्प का उपयोग करें। (कोड वारियर की विभिन्न रिलीज़ इस विकल्प के लिए अलग–अलग नेविगेशन मार्ग लेती हैं।) आपको प्रोसेसर के मोटोरोला 680x0 श्रृंखला के लिए 68KB संस्करण के बीच चयन करना पड़ सकता है),

एक पीपीसी संस्करण (पावरपीसी प्रोसेसर के लिए), या एक कार्बन संस्करण (ओएस एक्स के लिए)।

नई परियोजना में प्रारंभिक परियोजना के हिस्से के रूप में एक छोटा स्रोत कोड फ़ाइल

है। आप उस प्रोग्राम को कंपाइल और रन करने की कोशिश कर सकते हैं, यह देखने के लिए कि क्या आपका सिस्टम ठीक से सेट है।

LANGUAGE STANDARDS

वर्तमान में, कई सी कार्यान्वयन उपलब्ध हैं। आदर्श रूप से, जब आप एक सी प्रोग्राम लिखते हैं, तो उसे किसी भी कार्यान्वयन पर समान काम करना चाहिए, बशर्ते वह मशीन–विशिष्ट प्रोग्रामिंग का उपयोग न करे। व्यवहार में सही होने के लिए, अलग–अलग कार्यान्वयन को एक मान्यता प्राप्त standard के अनुरूप होना चाहिए।

पहले, सी के लिए कोई आधिकारिक standard नहीं था। इसके बजाय, ब्रायन कर्निघन और डेनिस विची द्वारा सी प्रोग्रामिंग भाषा का पहला संस्करण (1978) स्वीकृत standard बन गया, जिसे आमतौर पर K के रूप में जाना जाता है उदाहरण के लिए, कम्पाइलर एक पूर्ण के की पेशकश करने का दावा करेंगे हालाँकि, हालांकि इस परिशिष्ट ने सी भाषा को परिभाषित किया, लेकिन इसने सी लाइब्रेरी को परिभाषित नहीं किया। अधिकांश भाषाओं से अधिक, C इसकी लाइब्रेरी पर निर्भर करता है, इसलिए लाइब्रेरी standard की भी आवश्यकता है। किसी भी आधिकारिक standard की अनुपस्थिति में, UNIX कार्यान्वयन के साथ आपूर्ति की गई लाइब्रेरी एक वास्तविक standard बन गई।

THE FIRST ANSI/ISO C STANDARD

जैसा कि C विकसित और अधिक व्यापक रूप से विभिन्न प्रकार की प्रणालियों पर उपयोग किया गया, C समुदाय ने महसूस किया कि इसे और अधिक व्यापक, अद्यतित और कठोर standard की आवश्यकता है। इस आवश्यकता को पूरा करने के लिए, अमेरिकी राष्ट्रीय standard संस्थान (ANSI) ने 1983 में एक नए standard को विकसित करने के लिए एक समिति (X3J11) की स्थापना की, जिसे औपचारिक रूप से अपनाया गया था।1989. यह नया standard (ANSI C) भाषा और standard C लाइब्रेरी दोनों को परिभाषित करता है। standardised के लिए अंतर्राष्ट्रीय संगठन ने C standard (ISO C) को अपनाया1990. आईएसओ सी और एएनएसआई सी अनिवार्य रूप से एक ही standard हैं। ANSI / ISO standard के अंतिम संस्करण को अक्सर C90 कहा जाता है। हालांकि, क्योंकि एएनएसआई

संस्करण पहले बाहर आया था, लोग अक्सर एएनएसआई सी शब्द का इस्तेमाल करते थे।

समिति के कई मार्गदर्शक सिद्धांत थे। शायद सबसे दिलचस्प यह था: सी की रखें। समिति ने निम्नलिखित विचारों को उस Expressionsना के भाग के रूप में सूचीबद्ध किया:

> प्रोग्रामर पर भरोसा करें।
> प्रोग्रामर को ऐसा करने से न रोकें जो करने की आवश्यकता है।
> भाषा को छोटा और सरल रखें।
> ऑपरेशन करने के लिए केवल एक ही तरीका प्रदान करें।
> इसे तेज करें, भले ही यह पोर्टेबल होने की गारंटी न हो।

अंतिम बिंदु तक, समिति का अर्थ था कि एक कार्यान्वयन को एक विशेष ऑपरेशन को परिभाषित करना चाहिए जो एक सार, समान परिभाषा को लागू करने की कोशिश करने के बजाय लक्ष्य कंप्यूटर के लिए सबसे अच्छा काम करता है। भाषा सीखते ही आपको इस दर्शन के उदाहरण मिलेंगे।

C99 STANDARD

1994 में, standard को संशोधित करने पर काम शुरू हुआ, एक प्रयास जिसके परिणामस्वरूप C99 standard हुआ। एक संयुक्त ANSI / ISO समिति, जिसे तब C9X समिति के रूप में जाना जाता है, ने C90 standard के मूल सिद्धांतों का समर्थन किया, जिसमें भाषा को छोटा और सरल रखना शामिल है। उनका इरादा नए लक्ष्यों को पूरा करने के लिए भाषा को छोड़कर नई सुविधाओं को जोड़ना नहीं था। नए लक्ष्यों में से एक था, अंतर्राष्ट्रीय प्रोग्रामिंग का समर्थन करना, उदाहरण के लिए, अंतरराष्ट्रीय Character सेट से निपटने के तरीके प्रदान करना। दूसरा लक्ष्य "स्पष्ट कमियों को दूर करने के लिए मौजूदा अभ्यास को संहिताबद्ध करना था।" इस प्रकार, जब सी –64 बिट प्रोसेसर को स्थानांतरित करने की आवश्यकता को पूरा करते हैं, तो वे उन लोगों के अनुभवों पर standard के अतिरिक्त जोड़ देते हैं जो वास्तविक जीवन में इस समस्या से निपटते हैं। तीसरा लक्ष्य

वैज्ञानिक और इंजीनियरिंग परियोजनाओं के लिए महत्वपूर्ण संख्यात्मक गणना करने के लिए सी की उपयुक्तता में सुधार करना था।

इन तीन बिंदुओं का उपयोग, कमियों का सुधार, और कम्प्यूटेशनल उपयोगिता का सुधार main परिवर्तन oriented लक्ष्य थे। परिवर्तन की शेष योजनाएँ प्रकृति में अधिक रूढ ़िवादी थीं, उदाहरण के लिए, C90 के साथ असंगतताओं को कम करना और C के साथ और भाषा को वैचारिक रूप से सरल रखना। समिति के शब्दों में, "&" docEmphasis "> big और महत्वाकांक्षी

उत्थान यह है कि C99 परिवर्तन C की आवश्यक प्रकृति को संरक्षित करता है, और C एक दुबला, स्वच्छ, कुशल भाषा बनी हुई है। यह पुस्तक C99 के कई बदलावों की ओर इशारा करती है। क्योंकि इस समय अधिकांश कंपाइलर सभी C99 परिवर्तनों को पूरी तरह से लागू नहीं करते हैं, आप पा सकते हैं कि उनमें से कुछ आपके सिस्टम पर उपलब्ध नहीं हैं।

STANDARD NAMING CONVENTIONS

यह पुस्तक ISO / ANSI C शब्द का उपयोग नई विशेषताओं को संदर्भित करने के लिए standard और C99 दोनों के लिए सामान्य सुविधाओं का अर्थ करेगी। कभी–कभी, यह C90 को संदर्भित करेगा, उदाहरण के लिए, जब एक विशेषता को पहली बार C में जोड़ा गया था, तो चर्चा करना।

कुछ सम्मेलन

हम स्वयं सी भाषा का अध्ययन शुरू करने के लिए लगभग तैयार हैं। यह खंड प्रस्तुत सामग्री में हमारे द्वारा उपयोग किए जाने वाले कुछ सम्मेलनों को शामिल करता है।

टाइपफ़ेस

पाठ Programs और कंप्यूटर इनपुट और आउटपुट का प्रतिनिधित्व करने के लिए, हम एक प्रकार के फ़ॉन्ट का उपयोग करते हैं जो एक स्क्रीन पर या मुद्रित आउटपुट पर आपके द्वारा देखे जाने वाले जैसा दिखता है। हम पहले ही कुछ बार इसका इस्तेमाल कर चुके हैं। यदि यह आपके नोटिस को खिसका देता है, तो फॉन्ट निम्नलिखित की तरह दिखता है:

```
"Concrete contains gravel and cement.\n");
 return 0;
}
```

SCREEN OUTPUT

कंप्यूटर से आउटपुट एक ही प्रारूप में मुद्रित किया जाता है, इस अपवाद के साथ कि उपयोगकर्ता इनपुट बोल्डफेस प्रकार में दिखाया गया है।

वेब अध्ययन

सामान्य कंप्यूटर फ़ॉन्ट में मुद्रित लाइनें प्रोग्राम आउटपुट हैं, और बोल्डफेस लाइन उपयोगकर्ता इनपुट है।

```
Please enter the book title.
Press [enter] at the start of a line to stop.

My Life as a scary
Now enter the author.

Machhindra kalan
```

ऐसे कई तरीके हैं जिनसे आप और कंप्यूटर एक दूसरे से संवाद कर सकते हैं। हालाँकि, हम मान लेंगे कि आप कीबोर्ड का उपयोग करके कमांड टाइप करते हैं और आप स्क्रीन पर प्रतिक्रिया पढ़ते हैं।

विशेष कीस्ट्रोक्स

आमतौर पर, आप दर्ज की गई कुंजी, c / r, Return, या इनमें से कुछ भिन्नता को दबाकर निर्देशों की एक पंक्ति भेजते हैं। हम इस कुंजी को टेक्स्ट में एंटर कुंजी के रूप में संदर्भित करते हैं। प्रोग्राम उदाहरण में इसे इनपुट के भाग के रूप में दिखाते समय, हम [एंटर] का उपयोग करते हैं। brackets का अर्थ है कि आप शब्द दर्ज करने के बजाय एक ही कुंजी दबाते हैं।

हम नियंत्रण characters को भी संदर्भित करते हैं, जैसे कि Ctrl+d. यह संकेतन का अर्थ D कुंजी टाइप करने के दौरान है, जब आप कुंजी को दबाए हुए Ctl (या नियंत्रण) को दबाते हैं।

हमारा सिस्टम

C के कुछ पहलू, जैसे किसी संख्या को संग्रहीत करने के लिए उपयोग की जाने वाली जगह की मात्रा, सिस्टम पर निर्भर करती है। कब हम उदाहरण देते हैं और "हमारे सिस्टम" का संदर्भ लेते हैं, हम विंडोज 98 के तहत चलने वाले पेंटियम पीसी की बात करते हैं और Metrowerks CodeWarrior Pro 6.0, Microsoft Visual C ++ 6.0 का उपयोग कर, माइक्रोसॉफ्ट विजुअल सी6.0, जीसीसी2.95, या कोमू4.2.45। इस लेखन के समय, C99 समर्थन अधूरा है, और इनमें से कोई भी कंपाइलर सभी C99 सुविधाओं का समर्थन नहीं करता है। लेकिन, उनके बीच, जीसीसी, कोम्यू और कोडवर्ड नए standard के बहुत कवर करते हैं। अधिकांश उदाहरणों को एक Maciiosh G3 पर Metrowerks CodeWarrior Pro 6 का उपयोग करके भी परीक्षण किया गया है।

हम कभी–कभी UNIX प्रणाली पर चलने वाले Programs का भी उल्लेख करते हैं। हमारे द्वारा उपयोग किया जाने वाला बर्कले का बीएसडी हैएक वैक्स 11 / 750 कंप्यूटर पर चलने वाला UNIX का 4.3 संस्करण। इसके अलावा, कई Programs को पेंटियम पीसी पर लिनक्स

चलाने और जीसीसी2.95 का उपयोग करके परीक्षण किया गया था और कोमू4.2.45.

आपकी प्रणाली

आपको एक सी compiler या एक तक पहुंच की आवश्यकता है। C कंप्यूटर सिस्टम की एक विशाल विविधता पर चलता है, इसलिए आपके पास कई विकल्प हैं। सुनिश्चित करें कि आप अपने विशेष सिस्टम के लिए डिज़ाइन किए गए C कंपाइलर का उपयोग करते हैं। इस पुस्तक के कुछ उदाहरणों में नए C99 standard के लिए समर्थन की आवश्यकता है, लेकिन अधिकांश उदाहरण C90 compiler के साथ काम करेंगे। यदि आपके द्वारा उपयोग किया जाने वाला कंपाइलर प्री–एएनएसआई / आईएसओ है, तो आपको समायोजन करना होगा, शायद अक्सर आपको कुछ नया करने के लिए प्रोत्साहित करने के लिए पर्याप्त हो।

अधिकांश कंपाइलर विक्रेता छात्रों और शिक्षकों को विशेष मूल्य प्रदान करते हैं, इसलिए यदि आप उस श्रेणी में आते हैं, तो विक्रेता वेब साइटों की जांच करें।

SUMMARY

C एक शक्तिशाली, संक्षिप्त प्रोग्रामिंग भाषा है। यह लोकप्रिय है क्योंकि यह उपयोगी प्रोग्रामिंग टूल और हार्डवेयर पर अच्छा नियंत्रण प्रदान करता है और क्योंकि C प्रोग्राम एक सिस्टम से दूसरे सिस्टम में परिवहन के लिए सबसे आसान है।

सी एक compiled भाषा है। सी कंपाइलर और लिंकर्स ऐसे प्रोग्राम हैं जो C भाषा स्रोत कोड को निष्पादन योग्य कोड में बदलते हैं।

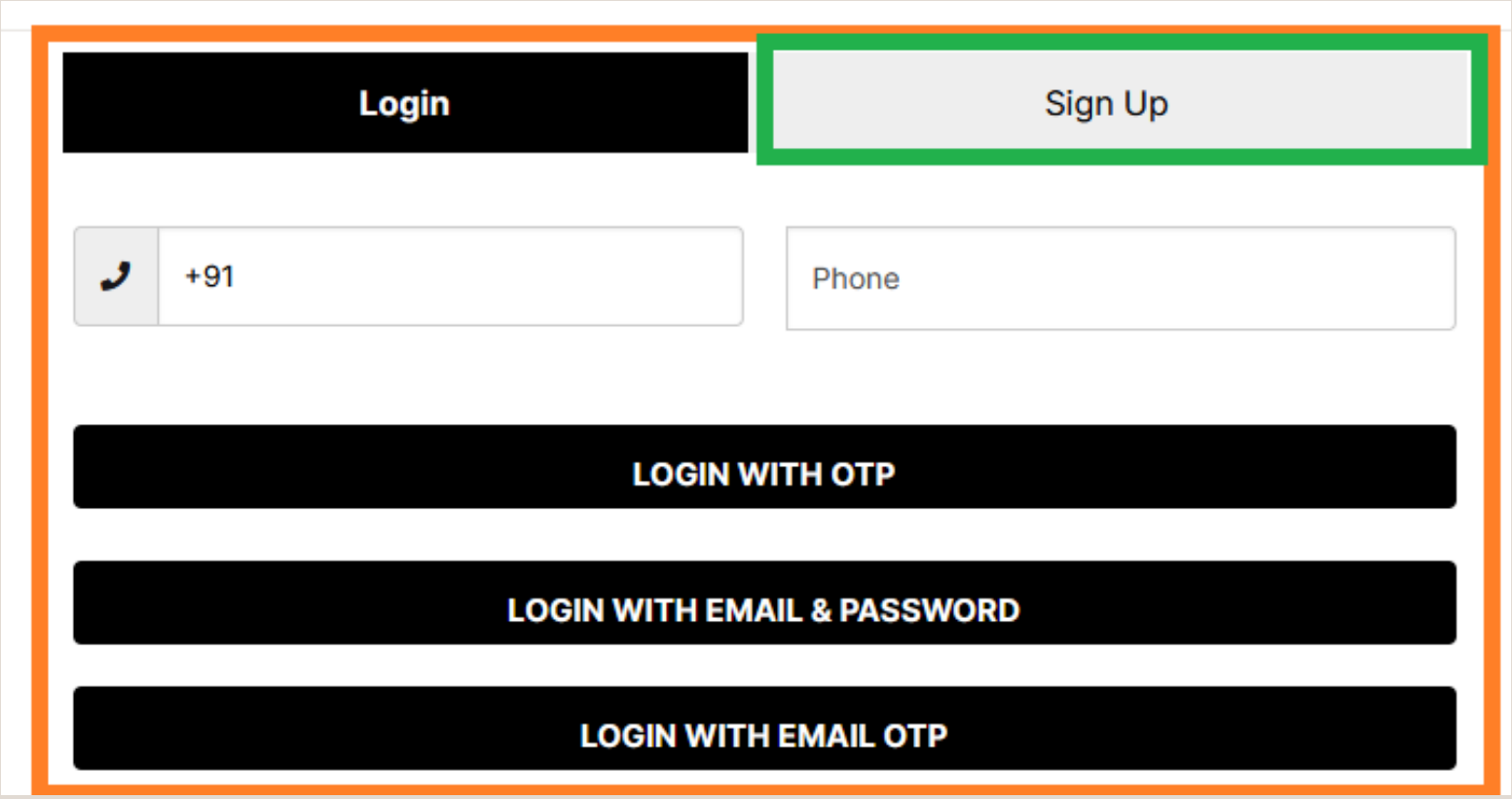
C में प्रोग्रामिंग कर योग्य, कठिन और निराशाजनक हो सकती है, लेकिन यह पेचीदा, रोमांचक और संतोषजनक भी हो सकती है। हमें उम्मीद है कि आप इसे उतना ही सुखद और आकर्षक पाएंगे जितना हम करते हैं।

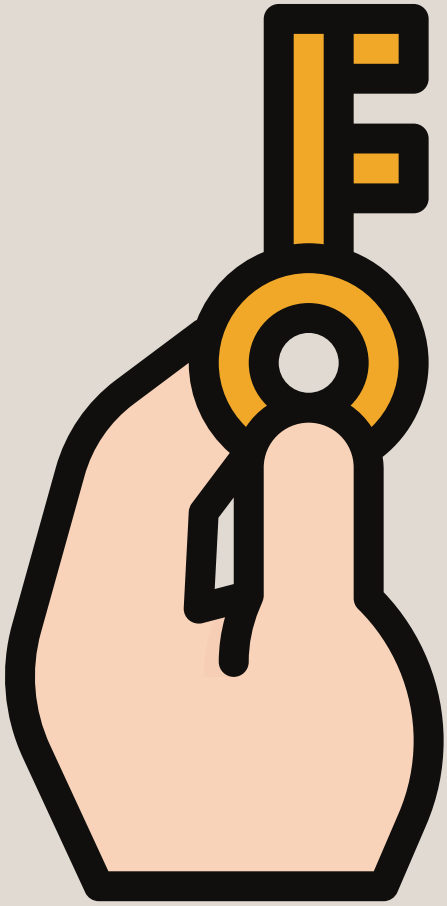
HOW TO GET COMPLETE PDF EBOOK

Pay Paytm

कोई भी **2 Ebooks** खरीदने पर **20%** डिस्काउंट- **India20**

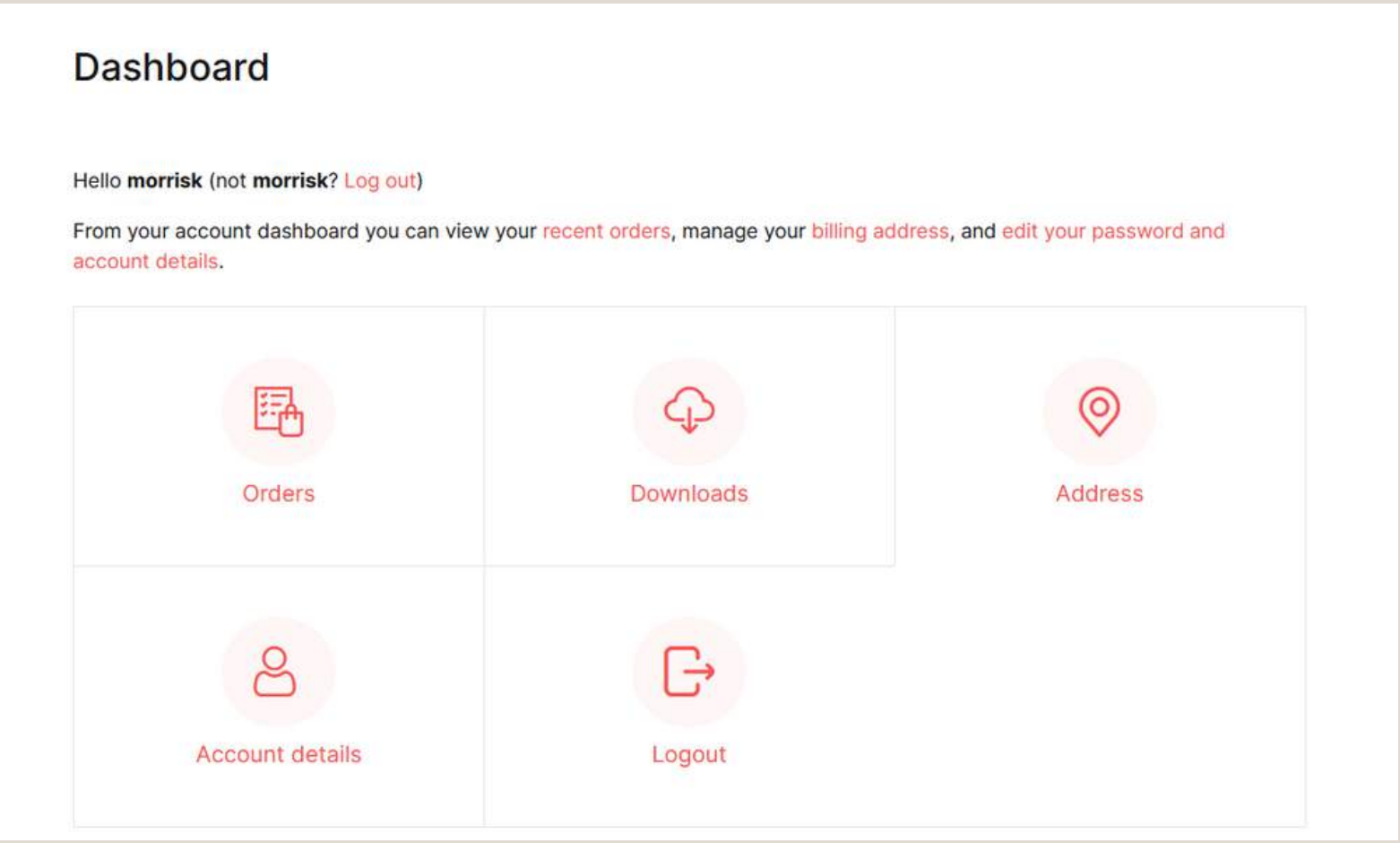
Buy Now पर क्लिक करने पर आपको **चेकआउट पेज** पर लॉगिन फॉर्म पर पहिले ईमेल से लॉगिन करना होगा।





Login Succesfull के बाद आपको पेमेंट करके Ebooks वही से डाउनलोड कर सकते है।

उसके बाद आप **My Account** पेज पर जाकर **Download** में Ebooks डाउनलोड कर सकते है।



Thank you